**FUNCTIONS CALL BY REFERENCE**

**AND RECURSIVE FUNCTIONS**

**LAB # 6**



**Spring 2019**

**CSE102L Computer Programming Lab**

Submitted by: **SHAH RAZA**

Registration No. : **18PWCSE1658**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr. Madiha Sher**

April 10, 2019

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

## Objectives:

* To understand the programming of recursive functions
* To understand function programming, its types and function-call

**TASK #1:**

Print 1 to 100 in C++ using recursion.

**Code:**

#include <iostream>

using namespace std;

int counter=0; //Global variable

int numbers() //Function definition

{

counter++; //Increment counter

cout<<counter<<" "; //Display counter

if(counter==100) //Function ends when counter=100

return 0;

else //Recursion

numbers();

}

int main()

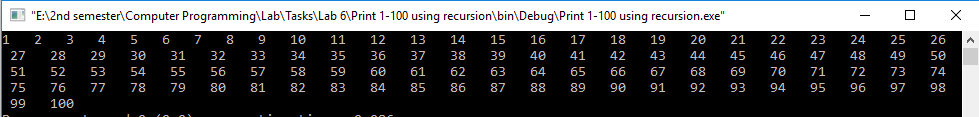
{

numbers(); //Function Call

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #2:**

Calculate the sum of odd natural numbers 1+3+5+7+……………. .+n using while loop. Take n as input from user.

**Code:**

#include <iostream>

using namespace std;

int sum(int N) //Function definition

{

static int i=1,Sum=0; //Variable declaration

Sum+=i;

i=i+2; //Increment by 2

if(i>N) //Function terminates when i>N

return Sum;

else //Recursion

sum(N);

}

int main()

{

/\*int num,i=1,sum=0;

cout<<"Enter a number: ";

cin>>num;

while(i<=num)

{

sum=sum+i;

i=i+2;

}

cout<<"Sum of odd natural numbers up to "<<num<<" is: "<<sum;\*/

int N,out; //Variable declaration

cout<<"Enter N: "; //Display message

cin>>N; //Input N

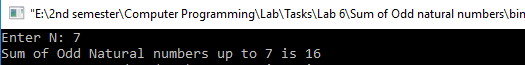
out=sum(N); //Function Call

cout<<"Sum of Odd Natural numbers up to "<<N<<" is "<<out; //Display sum

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #3:**

Write a program to swap value of two variables using function.

**Code:**

#include <iostream>

using namespace std;

void Swap(int &a, int &b) //Function definition

{

//Swapping Process

a=a+b;

b=a-b;

a=a-b;

}

int main()

{

int num1,num2; //Variable declaration

cout << "Number1: "; //Display message

cin>>num1; //Input num1

cout<<"Number2: "; //Display message

cin>>num2; //Input num2

Swap(num1,num2); //Function Call

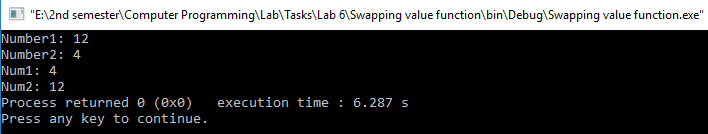
cout<<"Num1: "<<num1; //Display num1

cout<<"\nNum2: "<<num2; //Display num2

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #4:**

Write a function to find Sum of N natural numbers using Recursion.

**Code:**

#include <iostream>

using namespace std;

int sum(int N) //Function definition

{

static int i=1,Sum=0; //Variable declaration

Sum+=i;

i++; //Increment i

if(i>N) //Function terminates when i>N

return Sum;

else //Recursion

sum(N);

}

int main()

{

int N,out; //Variable declaration

cout<<"Enter N: "; //Display message

cin>>N; //Input N

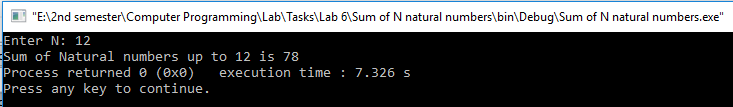
out=sum(N); //Function Call

cout<<"Sum of Natural numbers up to "<<N<<" is "<<out; //Display sum

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #5:**

Write a C++ Program to Find Factorial of a Number Using Recursion.

**Code:**

#include <iostream>

using namespace std;

int factorial(int num) //Function definition

{

if (num==0 || num==1) //Factorial of 0 and 1

return 1;

else //Recursion

{

int fact=num\*factorial(num-1);

return fact;

}

}

int main()

{

int num; //Variable declaration

cout<<"Enter a Number: "; //Display message

cin>>num; //Input num

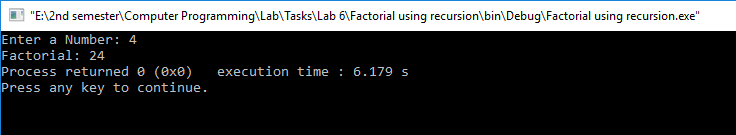
int out=factorial(num); //Function Call

cout<<"Factorial: "<<out; //Display Factorial

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #6:**

C++ Program to Find G.C.D Using Recursion.

**Code:**

#include <iostream>

using namespace std;

int GCD(int a, int b) //Function definition

{

//Finding GCD

if(a>b)

a=a-b;

if(b>a)

b=b-a;

if(a==b)

return a;

else //Recursion

GCD(a,b);

}

int main()

{

int num1,num2,out; //Variable declaration

cout << "Enter Num1: "; //Display message

cin>>num1; //Input num1

cout<<"Enter Num2: "; //Display message

cin>>num2; //Input num2

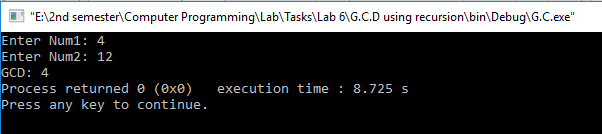
out=GCD(num1,num2); //Function Call

cout<<"GCD: "<<out; //Display GCD

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #7:**

C++ program to print fibonacci series using recursion.

**Code:**

#include <iostream>

using namespace std;

int fibonacci(int N) //Function definition

{

static int a=0,b=1; //Variable declaration

if(a==0)

cout<<a<<" "<<b<<" ";

int c=a+b; //Variable declaration

a=b;

b=c;

cout<<c<<" "; //Display c

if(c>N)

return 0;

else //Recursion

fibonacci(N);

}

int main()

{

int num; //Variable declaration

cout<<"Enter a number: "; //Display message

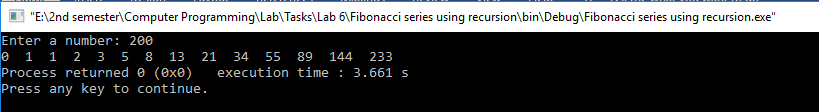
cin>>num; //Input num

fibonacci(num); //Function Call

return 0;

}

**Output (Compilation, Debugging and Testing):**



**TASK #8:**

C++ program to calculate power of a number using recursion.

**Code:**

#include <iostream>

using namespace std;

int pow (int base,int exponent) //Function definition

{

static int power=1; //Variable declaration

power=power\*base;

exponent--; //Decrement exponent

if(exponent==0) //Function terminates when exponent is 0

return power;

else //Recursion

pow(base,exponent);

}

int main()

{

int base,exp; //Variable declaration

cout<<"Base: "; //Display message

cin>>base; //Input base

cout<<"Exponent: "; //Display message

cin>>exp; //Input exp

int out=pow(base,exp); //Function Call

cout<<"Result: "<<out; //Display Power

return 0;

}

**Output (Compilation, Debugging and Testing):**

